

РОЗДІЛ 4

АВТОМАТИКА, КОМП'ЮТЕРНІ
ТА ТЕЛЕКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

УДК 621.564

М. Г. Хмельнюк✉, *Д. І. Важинський*

Одесская национальная академия пищевых технологий, ул. Канатная, 112, Одесса, 65039, Украина

✉ e-mail: hmel_m@ukr.net

АЛГОРИТМИЗАЦИЯ ТЕРМОДИНАМИЧЕСКИХ РАСЧЕТОВ В МАТЕМАТИЧЕСКОМ ПАКЕТЕ
MAPLE С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ OPENMAPLE

В статье описывается создание динамической библиотеки, включающей в себя функции для работы с идеальным газом. В частности, рассматривается обработка цикла, состоящего из произвольного количества точек. Показаны особенности алгоритмизации термодинамических расчетов. Предложен алгоритм обработки простейшего цикла без смешения. При создании библиотек использовалась технология OpenMaple.

Ключевые слова: Идеальный газ; Maple; OpenMaple.

М. Г. Хмельнюк, Д. І. Важинський

Одеська національна академія харчових технологій, вул. Канатна, 112, Одеса, 65039, Україна

АЛГОРИТМІЗАЦІЯ ТЕРМОДИНАМІЧНИХ РОЗРАХУНКІВ В МАТЕМАТИЧНОМУ ПАКЕТІ
MAPLE З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ OPENMAPLE

В статті має місце опис створення динамічної бібліотеки, яка включає в себе функції для роботи з ідеальним газом. Показані особливості алгоритмізації термодинамічних розрахунків. Розглянута обробка циклу, який включає в себе довільну кількість точок. Запропоновано алгоритм обробки найпростішого циклу без змішування. При створенні бібліотеки використовувалася технологія OpenMaple.

Ключові слова: Ідеальний газ; Maple; OpenMaple.



This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>

I. ВВЕДЕНИЕ

При написании собственных программ автоматизированного проектирования (САПР) одной из важных проблем является проблема алгоритмизации термодинамических циклов. При программировании таких задач как цикл состоящий из некоторого нефиксированного количества точек и процессов нет возможности провести расчеты по некоторой определенной методике. Поэтому появляется необходимость в создании определенного подхода, который учитывал бы максимальное количество ситуаций. Любой термодинамический цикл можно представить, как совокупность точек и процессов. Любые данные характеризуют либо процесс, либо саму точку. Вообще любой обратимый термодинамический цикл можно представить в виде неориентированного графа, состоящего из узлов и связей [1]. Данный подход упрощает понимание цикла и позволяет использовать преимущества теории графов.

Уравнения, характеризующие цикл, также можно разделить на уравнения состояния, которые харак-

теризуют узлы, либо на уравнения процесса характеризующие связи. Данный подход позволяет разделить все процедуры и функции – одни обрабатывают узлы, другие работают со связями.

II. ПОСТАНОВКА ЗАДАЧИ

Первым этапом, при разработке любого кода является правильная постановка задачи. Основными задачами, помимо непосредственных расчетов, является максимальная автоматизация и удобство ввода начальных данных. Результаты, получаемые в математическом пакете должны быть пригодны для дальнейших расчетов в зависимости от ситуации.

Одной из наиболее простых математических моделей является модель идеального газа. Формализация данной задачи выражается в использовании ряда зависимостей:

уравнения Менделеева–Клапейрона

$$pv = RT; \quad (1)$$

уравнения политропы

$$pv^n = \text{const.} \quad (2)$$

Остальные термодинамические величины могут быть легко рассчитаны на основе вышеизложенных формул и первого закона термодинамики:

$$dq = du + pdv. \quad (3)$$

Также стоит учесть, что при расчетах стоит избегать деления на ноль и умножения на бесконечность.

При разработке полноценной библиотеки оказывается, что ее размер может превышать две тысячи строк, а некоторые задачи, которые при этом необходимо дополнительно решать, носят далеко не физический характер. Это вызвано необходимостью обработки исключительных ситуаций, а также зависит от функций, которые будут в ней заложены.

III. ПРОСТЕЙШАЯ ПРОЦЕДУРА

В разработанной динамической библиотеке предполагается доопределение одной термодинамической точки, возможность построения цикла, а также вычисление некоторых функций процесса, например, работы сжатия.

Наиболее простой задачей, может быть доопределение точки. Данная функция затем будет применяться при анализе цикла. Например, при известном давлении и температуре, необходимо рассчитать удельный объем воздуха. В данном случае в Maple данная процедура будет выглядеть следующим образом:

```

C1 := Array([12.61, 700, NOT, 28.9])
[ 12.61 700 NOT 28.9 ]
evalf4(Simple1(C1))
Absolute pressure = 12.610000 Bar
Absolute temperature = 700.000000 K
Specific volume = 0.159696 m3/kg
Molar mass = 28.900000 g/mol
[ 12.61 700. 0.1597 28.90 ]

```

Стоит отметить, что массив после обработки меняет свои значения, и может быть использован для дальнейших вычислений, которые не обязательно программировать. После окончания разработки функции, ее рекомендуется задокументировать [2]. Пользователь не знает, на какой позиции в массиве должно находиться давление или же что введение показателя политропы процесса, равного 1, будет эквивалентно введению текстового значения T. При программировании таких функций всегда присутствует некоторая методика, детали которой могут быть со временем забыты, что может быть проблематичным при использовании ее другими людьми.

IV. ТЕРМОДИНАМИЧЕСКИЙ ЦИКЛ С ПРОИЗВОЛЬНЫМ КОЛИЧЕСТВОМ ТОЧЕК И ПРОЦЕССОВ

Если расчет давления при имеющейся температуре и заданном удельном объеме не является сложной задачей, то построение цикла, состоящего из n -го количества точек при заданных характеристиках процесса (изобарный, политропический и т.д.), является не такой простой задачей, как может показаться на первый взгляд. Блок-схема данной процедуры представлена на рисунке 1.

Из вышеописанной блок-схемы видно, что, кроме непосредственного расчета по введенным данным, требуется решение вопроса согласования типов данных, решение задачи переопределенности, а также анализ данных на простую ошибку при вводе данных пользователем. Предположим, вводятся две точки, параметры которых могут быть рассчитаны независимо друг от друга, однако при этом, введено значение политропы процесса 1.1. В данном случае нет необходимости описывать процесс, либо стоит ввести меньше данных при описании точки. Однако, если по ошибке вводится что-то лишнее, и при этом противоречает предыдущим данным, программа должна сообщить об этом пользователю. С другой стороны, всегда существует вероятность неправильного введения данных пользователем, например, в функцию может быть по ошибке подставлен массив не той размерности. В данном случае последствия работы могут быть непредсказуемыми – от простого закрытия программы до зависания компьютера. Решение вышеупомянутых проблем стоит объединить в одну процедуру, которая будет возвращать логическое «да» или «нет» в зависимости от качества данных. С практической точки зрения, для решения большинства задач, связанных с идеальным газом, разработать стоит лишь несколько функций, которые, тем не менее, будут значительными по размеру, и будут включать в себя другие подпрограммы, которые для удобства будут разработаны отдельно и не будут доступны пользователю. Например, процедура, рассчитывающая давление при известной температуре и удельном объеме при заданном типе процесса. В Maple применение вышеописанной процедуры будет выглядеть следующим образом.

```

A := Array([4, NOT, NOT, 28.7])
[ 4. NOT NOT 28.7 ]
B := Array([5, 332, NOT, NOT])
[ 5. 332. NOT NOT ]
C := Array([8, NOT, NOT, NOT])
[ 8. NOT NOT NOT ]
evalf4(Simple3(A, 1.4, B, 1.4, C)) :
[ 4. 311.5 0.2256 28.70 ]
[ 5. 332. 0.1924 28.70 ]
[ 8. 379.7 0.1375 28.70 ]

```

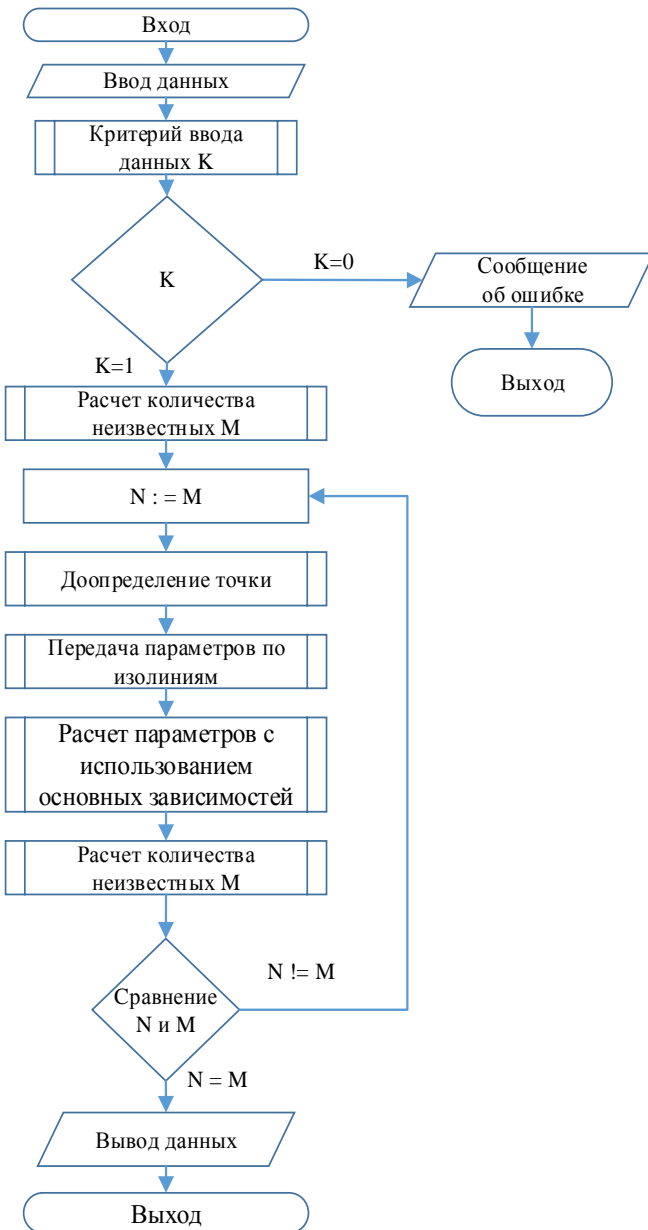


Рисунок 1 – Блок схема расчета.

Вначале пользователь вводит данные, которые у него есть. В данном случае предполагается использование массивов, состоящих из четырех точек. В случае если некоторые параметр неизвестен, с клавиатуры вводится значение NOT. Тип данных, которые хранятся в массиве, будут **anything** [3]. В алгоритме обработки массива предусматривается распознавание неизвестной в том случае, если данный параметр имеет текстовый тип, то есть тип **string**. Потому, в принципе, не важно, какое слово будет использоваться, главное, чтобы неизвестный элемент массива был

заполнен буквами. При построении процессов, данные вводятся по следующим соображениям, сначала в функцию, вводится начальная точка, в данном случае точка A, затем вводится характеристика процесса, это может быть, значение полтропы, в данном случае 1.4, что свидетельствует о том, что процесс адиабатный. Так же, вместо числа может быть введен некоторый буквенный параметр, например, T, что означает, что процесс изотермический. Далее вводится точка B, и процесс завершается. После этого снова можно ввести характеристику процесса, и завершить его точкой. При этом, не всегда может быть достаточно введенных данных до полного доопределение точки, однако это не всегда и требуется, поэтому параметров точек, которые так и не могли быть рассчитаны, будут и далее хранить текстовую информацию, которая была введена при создании точки. В вышеуказанном примере было построено адиабатное сжатие в компрессоре, с 4 бар до 8. При этом от пользователя не требуется производить каких-либо расчетов, основная задача сводиться к корректному вводу данных и правильной постановке задачи.

V. ВЫВОДЫ

При алгоритмизации расчетов термодинамических циклов внимание должно быть уделено не только формализации задачи, но и формату хранения данных и четкой классификации необходимых процедур [4]. Не менее важно правильно определиться с форматом ввода данных. На данном примере четко видно, что нет возможность задать точки в которых происходит смешивание потоков. Описанный подход вполне применим при отсутствии точек смешения, в случае же более сложных циклов необходимо использовать теорию графов и матрицу связей, которая позволяет зафиксировать данные в более удобной форме.

ЛИТЕРАТУРА

1. Зыков А. А. Основы теории графов. – М.: «Вузовская книга», 2004. — С. 664. — ISBN 5-9502-0057-8. (М.: Наука, 1987. 383 с.)
2. MaplePrimes [Электронный ресурс]. Режим доступа: <http://www.mapleprimes.com>. Дата обращения: 16.06.2016.
3. Maplesoft [Электронный ресурс]. Режим доступа: <http://www.maplesoft.com> Дата обращения: 16.06.2016,
4. Брайан Керниган, Деннис Ритчи. Язык программирования Си. — Санкт-Петербург: Невский диалект, 2001. — 352 с. — (Библиотека программиста). — ISBN 5794000457.

Отримана в редакції 11.05.2016, прийнята до друку 01.07.2016

M. G. Khmelniuk[✉], **D. I. Vazhinsky**

Odessa National Academy of Food Technologies, 112 Kanatnaya str., Odessa, 65039, Ukraine

✉ e-mail: hmel_m@ukr.net

ALGORITHMIZATION OF THERMODYNAMIC CALCULATIONS IN MAPLE MATHEMATICAL PACKAGES USING OPENMAPLE TECHNOLOGY

This article describes creating of dynamic link library, which includes functions for working with the ideal gas. In particular, thermodynamic cycle is analysed, which consists of arbitrary quantity of points. Features of thermodynamic cycles algorithmization are shown. An algorithm of simplest cycle without mixing is proposed. For creating library, the OpenMaple technology was used. The article deals with the thermodynamic cycles algorithmization, it describes cycle calculation, which has arbitrary quantity of points. The main problem in algorithmization random cycle is lack of data of the number of points and processes. In order to solve the problem of the lack of data it is necessary to develop a universal method that allows to abstract from the formulas. One option is to present the cycle in the form of a graph. This allows to formalize graph easier. Introduction of nodes and links allows to develop the necessary functions and procedures easily. An algorithm for calculating the cycle with an arbitrary number of points without mixing was proposed. The dynamic library, which can be connected to the package Maple was created. The main difficulty in writing the procedures is the large capacity of the code, a significant number of service functions and exception handling. It is worth paying attention to the harmonization of data types, since their mismatch can lead to freezing the computer and other exceptional situations. An important question is cycle predetermination.

Keywords: Ideal Gas; Maple; OpenMaple.

REFERENCES

1. **Zykov, A. A.** (2004). Osnovy teorii grafov. Moskow «Vuzovskaja kniga», 664 p. ISBN 5-9502-0057-8 (in Russian)
2. MaplePrimes [Electronic source] Available at: <http://www.mapleprimes.com> (Access date: 16 June 2016)
3. Maplesoft [Electronic source] Available at: <http://www.maplesoft.com> (Access date: 16 June 2016)
4. **Brajan Kernigan, Dennis Ritchi** (2001). Jazyk programirovaniia C. Sankt-Peterburg: Nevskij dialekt, 352 p. ISBN 5794000457 (in Russian).

Received 11 May 2016
Approved 01 July 2016
Available in Internet 31 August 2016